Iterable Systems

By Thomas Park

As I did my work of selecting sounds to use, for creating generative/iterative pieces, I ran across a particular type of work. I call this work an “Iterable System”.

In an Iterable System, the piece or pieces can be broken into small fragments and they will sound musical when the fragments are recombined. Some albums are like this, such as Brian Eno’s “The Shutov Assembly” (Warner Bros., 1992). Sections of any track from this album can be combined with other fragments or pieces.

This suggests several things. The musician who composed this type of system did so consciously, I would think, and planned for certain tunings, scales and modes before or while composing. In other words, the composer established rules for the system and abided by them.

As a result, their work or works form a particular type of system I call “iterable”.

Many older works, from before the 20th Century, are not iterable. This is because they are episodic in structure. They tend to tell a story, or stories-- and the dramatic storyline becomes the main musical rule. Possible modal restrictions are broken in order to tell the music’s story. Tchaikovsky’s “Nutcracker Suite” is likely not iterable, where Satie’s “Gymnopedies” are.

Iterable systems are ideal for generative work, as they can be broken into pieces and re-assembled in real-time in any combination. They make it possible for an entire album, or even albums of material, to recombine on top of one another.

Ideally I will find many cc-licensed systems that are iterable. In the meantime, as an exercise, I have been testing copywritten recordings, in the privacy of my own studio, to provide examples of this type of system, and to demonstrate their aesthetic qualities. Often, when my ear suggests that a work or works are iterable, I am correct. But not always. Sometimes they turn out to be narrative, overly simple or plain dissonant.

Iterable music works are perhaps aligned with iterable data sets-- these are sets of data in which part of the data reflects the whole-- fractal or recursive data, for example. These sets use the same types of measurements and establish predictable intervals throughout, as defined by the group of strict rules used to collect and map out the data.

The kind of consistency embodied by iterable systems results in that they are very useful for being simply recombined into multiple new versions, iterations, and combinations.